Gaetano Manzo

Online Human Assisted and Cooperative Pose Estimation of 2D-cameras

MASTER THESIS
Supervised by Prof. Francesc Serratosa & Prof. Mario Vento

Department of Computer Engineering and Mathematics
Index

Introduction ......................................................................................................................................... 1

1. State of the art and basic methods .................................................................................................. 4
   1.1. Salient point extractor and feature matching ............................................................................ 5
   1.2. Structure from motion and human interactivity ........................................................................... 7

2. The proposed model .......................................................................................................................... 10
   2.1. Human-assisted matching estimator ......................................................................................... 13
   2.2. Cooperative pose estimator ........................................................................................................ 16

3. Practical Validation ............................................................................................................................ 21
   3.1. First analysis: Camera positioning on open spaces ................................................................. 22
   3.2. Application 1: Robotics positioning ............................................................................................ 25
   3.3. Application 2: Camera calibration .............................................................................................. 27
       3.3.1. Images acquisition .............................................................................................................. 28
       3.3.2. Practical specifics ............................................................................................................... 30
       3.3.3. Salerno Lab Database ....................................................................................................... 32
   3.4. User Simulation ......................................................................................................................... 34

Conclusion ............................................................................................................................................ 40

Figures .................................................................................................................................................. 42

Tables .................................................................................................................................................... 44

Bibliography .......................................................................................................................................... 45
Introduction

In the course of this thesis we want to present a method to obtain the pose and orientation of several 2D cameras\(^1\) that has two novel features.

- The first one is the human can assist in mapping salient points between pairs of images when one estimates the error is too high\(^2\).
- The second one is the ability of the method to deduct the poses in a cooperative manner.

The system always tries to automatically deduct the poses and orientation of the whole cameras meanwhile the human visualises the deducted poses together with the estimated errors\(^3\) in a human-machine interface but, when the human considers appropriate, one can asynchronously interact on the system. Thus, the supervisor can partially modify the point-to-point mapping between two images, which increases the quality of the deducted homography between these images. This interaction makes to decrease not only the poses error of the involved two cameras but also the pose errors of the whole set of cameras.

In robotics or camera surveillance, properly estimating the pose and orientation of cameras is considered to be a crucial low-level task. In the first case, cameras are embedded on the robots and therefore their positions are mobile. Contrarily, in the second case, cameras are usually static. Nevertheless, in both cases, without properly setting the camera poses\(^4\), the method is not able to

---

\(^1\) Sometimes in the document, we refer to the robots as meaning cameras and conversely.

\(^2\) An alternative could be a new human’s mapping where through it, the system can update its mapping.

\(^3\) In this particular case, we deduct the estimate error through the key points mapped. Forward we deduct the estimate error without consider the key points but only the camera’s homography.

\(^4\) Orientation’s value difference should not be much between consecutive cameras.
deduct the position, direction, speed or acceleration of the objects or humans that are in the surroundings.

The method we present is part of a larger project in which social robots guide people through urban areas (http://www.iri.upc.edu/project/show/144). Previously it has been analysed the relation between humans and robots and also the behaviour of humans when social robots move closer to them (GARRELL & SANFELIU, 2012). In addition, it has been presented a tracking method that follows people, which allows occlusions and mobile cameras (SERRATOSA, ALQUÉZAR & AMÉZQUITA, 2012) and a robot navigation (FERRER, G., SANFELIU, A., 2014). Moreover, we have presented some results on structure from motion. That is, given several 2D cameras, the method reconstructs the 3D position of the cameras (RUBIO ET. AL., 2015). Finally, two other papers have been presented with respect to the human interaction. In the first one, the homography between 2D images is computed (CORTÉS & SERRATOSA, 2015) and in the second one, the 3D positions of the robots are deducted given 3D cameras (CORTÉS & SERRATOSA, 2016). In this paper, we move one step farther since robots only have 2D cameras and the method obtains the 3D pose of the cameras. Moreover, the user can interact asynchronously with the method to decrease the pose error. Note several levels of interaction could be considered. The highest level could be to impose the position of a camera due to this knowledge has been acquired through another method. Our proposal is related with the lowest level. A human is very good and fast at mapping points on two different scenes, independently of the intrinsic or extrinsic characteristics of the images. Thus, what is asked the user to do is simply to select a salient point on one of the images and map this point on another image. Note this action is performed asynchronously to the process of deducting the pose and the supervisor tends to perform it when robots has to stop due to the system is not able to deduct

\[5\] In the sense that the human easily matching the right points meanwhile a computer matching thousand point that can be wrong.
the pose in a completely automatic way. Therefore, the human interaction is a mechanism to make possible the robot asks to continue in extreme situations. The rest of the thesis is organized as follows. In the first chapter, we begin to summarise the state of the art related on human-robot interaction. Then, we explain how to deduct the relative pose of an object with respect to another one given an affine 3D homography. We also comment four basic methods used in our robotic system, which are salient point extractor, feature matching, structure from motion and human interaction. In chapter 2, we present our model. We first describe the main scheme and then we concretise on two specific parts: the human assisted matching estimator and the cooperative pose estimator. In the last chapter, we experimentally validate our model. First we show that with few human interactions, the accuracy of the estimated pose drastically increases. Then, we apply our method on automatic robot positioning and camera calibration. Note the human-machine interface has been described in the last part of this thesis.

_Gaetano Manzo_

gaetanomanzo@gmail.com
Chapter 1

Truth is ever to be found in simplicity, and not in the multiplicity and confusion of things.
Isaac Newton

1. State of the art and basic methods

In recent years, interaction between robots and humans and also cooperation between robots has increased rapidly. Applications of this field are very diverse, ranging from developing automatic exploration sites (TREVAI, Ota, FUKAZAWA, YUASA, ARAI & ASAMA, 2004) to using robot formations to transport and evacuate people in emergency situations (CASPER & MURPHY, 2003), assembly lines (UNHELKAR, SHAH, 2015) or simply vehicle positioning (IFTHEKHAR, SAHA, JANG, 2015). Within the area of social and cooperative robots we have (KIM, TAGUCHI, HONG & LEE, 2014; GARCIA, CENA, CARDENAS, SALTAREN, PUGLISI & SANTONJA, 2013) and in hospital care application we have (JEONG ET AL., 2015). As commented in the introduction, interactions between a group of people and a set of accompanying robots have become a primary point of interest in (GARRELL & SANFELIU, 2012). These methodologies, which are involved on pose estimation, usually assume the transformation between two 3D images or two sets of 3D points is modelled as an affine transformation in the 3D space. Then homography $H_{i,j}$ is defined as follows,

$$H_{i,j} = \begin{bmatrix} a_{i,j} & 0 & 0 & x_{i,j} \\ 0 & b_{i,j} & 0 & y_{i,j} \\ 0 & 0 & c_{i,j} & z_{i,j} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $a_{i,j} = S_{i,j} \cdot \cos(\beta_{i,j}) \cdot \cos(\gamma_{i,j})$, $b_{i,j} = S_{i,j} \cdot \cos(\alpha_{i,j}) \cdot \cos(\gamma_{i,j})$ and $c_{i,j} = S_{i,j} \cdot \cos(\alpha_{i,j}) \cdot \cos(\beta_{i,j})$. Parameter $S_{i,j}$ is the scale and $\alpha_{i,j}$, $\beta_{i,j}$ and $\gamma_{i,j}$ are the three orientation angles of one robot with respect to the other.
Besides $x_{i,j}$, $y_{i,j}$ and $z_{i,j}$ is the translation of one robot with respect to the other. Thus, given the affine homography $H_{i,j}$, the relative positions are simply values $x_{i,j}$, $y_{i,j}$ and $z_{i,j}$. Moreover, one combination of $\alpha_{i,j}$, $\beta_{i,j}$ and $\gamma_{i,j}$ angles is deducted solving the following equations,

\[
\cos(\beta_{i,j}) \cdot \cos(\gamma_{i,j}) = \frac{a_{i,j}}{S_{i,j}}
\]

\[
\cos(\alpha_{i,j}) \cdot \cos(\gamma_{i,j}) = \frac{b_{i,j}}{S_{i,j}}
\]

\[
\cos(\alpha_{i,j}) \cdot \cos(\beta_{i,j}) = \frac{c_{i,j}}{S_{i,j}}
\]

Errors in poses are considered as described in (Huynh, D.Q., 2009). The position error is computed as the Euclidean distance between the computed position in 3D and the ground truth position. The orientation error is computed as \[\|I - H_{i,j}^{sys} \cdot (H_{i,j}^{grt})^T\|_F / 2\sqrt{2}\] where $\|.\|_F$ is the Frobenius norm, $H_{i,j}^{sys}$ and $H_{i,j}^{grt}$ are the obtained and ground truth homographies between images $i$ and $j$ and $(.)^T$ represents the transpose matrix. The range of this expression is $[0,1]$ since it is demonstrated in (Huynh, D.Q., 2009) that the maximum value of $\|I - H_{i,j}^{sys} \cdot (H_{i,j}^{grt})^T\|_F$ is $2\sqrt{2}$.

### 1.1. Salient point extractor and feature matching

Salient points are image locations that can be robustly detected among different instances of the same scene with varying imaging conditions and they play the role of parts of the image to be matched. The most used local features to detect these points are:

---

6 An alternative is to replace the Frobenius norm above by the 2-norm to reduce the range of values to $[0, 2]$ instead $[0,2\sqrt{2}]$. 

• FAST (ROSTEN, E., Reid Porter, R., DRUMMOND, T., 2010): It is composed of a vector of features obtained from an algorithm that the authors call the accelerated segment test algorithm. This algorithm uses an approximation metric to determine the corners of an image.

• HARRIS (HARRIS, C., STEPHENS, M., 1988): It is composed of a vector of features obtained from the Harris Stephens algorithm. It is able to find corners and edges based on a local auto-correlation function.

• MINEIGEN (JOLLIFFE, I.T., 2002): It is composed of a vector of features obtained from the minimum eigenvalue algorithm. This algorithm determines the location of the corners based on the eigenvector and eigenvalue domain. It is originally designed for tracking purposes and it is able to detect some occlusions in the image.

• SURF (BAY, H., Ess, A., TUYTELAARS, T., VAN GOOL, L., 2008): It is composed of a vector of features obtained from the Speeded-up robust features algorithm. It is able to detect multiscale objects (known as blobs) as well as scale and rotation changes.

• SIFT (LOWE, D.G., 2004): It is composed of a vector of features obtained from the scale-invariant feature transform algorithm. It applies the Gaussian difference given several regions of the image in order to find scale and rotation features.

There is an evaluation of the most competent approaches in (MIKOLAJCZYK & SCHMID, 2005). When salient points have been detected, several correspondence methods can be applied that obtain the alignment (or homography) that maps one image into the other (ZHANG, Z., 1994), discards outlier points (FISCHLER & BOLLES, 1981) or characterises the image into an attributed graph (SANROMÀ, ET AL., 2012A; SANROMÀ, ET AL., 2012B; SERRATOSA & CORTÉS, 2015; SERRATOSA, 2014; SERRATOSA, 2015A; SERRATOSA, 2015B; SOLÉ, ET AL., 2012). Typically, these methods have been applied on 2D images but recently, 3D shape retrieval methods have appeared (LIA, B., 2015).
Some correspondence methods consider a rigid deformation from one image to the other one and other ones consider a non-rigid deformation. In the first case, it is assumed the whole image (and so, the extracted salient points) suffers from the same deformation and so the image alignment parameters are applied equally to the whole salient points or image pixels. Some examples are (Sanromà, et al., 2012a; Luo & Hancock, 2003; Rangarajan, Chui & Bookstein, 1997; Gold, Rangarajan, 1996). In the second case, each salient points\(^7\) suffers a different projection and there are different alignment parameters applied to each salient point or image region. Some examples are (Myronenko, Song, 2010; Chui, Rangarajan, 2003). Usually, the rigid strategy is applied to detect objects on outdoor images in which the deformation is mostly due to the change of the point of view. The non-rigid strategy is mostly applied to object detection or matching in medical or industrial images due to it is assumed objects suffer from deformations although the point of view is the same.

1.2. Structure from motion and human interactivity

Structure from motion refers to the process of estimating 3D structures from 2D image sequences (Xu, C., Tao, D., and Xu, C., 2015), which may be coupled with local image features. Finding structure from motion presents a similar problem to finding structure from stereovision. In both instances, the correspondence between images and the reconstruction of 3D object needs to be found. To find the correspondence between images, local features commented in the previous sub-section are tracked from one image to the next.

There are several approaches to solve structure from motion (Yi, G., Jianxin, L., Hangping, Q., Bo, W., 2014; Gui, J., Tao, D., Sun, Z., Luo, Y., You, X., and Tang, Y. 2014). In incremental structure from motion, poses are solved by adding one by one to the collection. In global structure from motion, the poses of all

---

\(^7\) It is not necessary that each point suffers a different deformation but also clusters of them (group of points with same features).
cameras are solved at the same time. A somewhat intermediate approach is out-of-core structure from motion, where several partial reconstructions are computed that are then integrated into a global solution.

Humans are very good at finding the correspondences between local parts of an image regardless of the intrinsic or extrinsic characteristics of the point of view. Human interactivity on feature matching has been applied on medical images (Pfluger & Thomas, et al., 2000; Pietrzyk, et al., 1994; Khader & Ben Hamza, 2012). Moreover, two patents have been presented (Jens Von & Neitzel, 2010; Gering & David, 2010). These methods are specific on some medical environments and for this reason cannot be applied on our problem. In (Pfluger & Thomas, et al., 2000), they show a comparison of 3D images on MRI-SPECT format and they concretise on images from the brain. In (Pietrzyk, et al., 1994), authors present a method to validate the 3D position given 3D medical images. Finally, in (Khader & Ben Hamza, 2012), the aim is to solve the feature matching given similar medical images extracted from different sensors or technologies. Patent (Von & Neitzel, 2010) defines a system for registration thorax X-Ray images such that it does not depend on bony structures. Finally, patent (Gering & David T., 2010) defines a multi-scale registration for medical images where images are first aligned at a course resolution, and subsequently at progressively finer resolutions; user input is applied at the current scale. Another usual application of human interaction is semi-automatic video annotation (Bianco, Ciocca, Napoletano & Schettini, 2015).

The method we present in this paper puts in common structure from motion and human interactivity. Given a correct (or almost correct) salient point correspondence between some images, completely automatic methods are really good at deducting the 3D pose of cameras however it is a difficult task for

---

This approach usually is well performed in term of temporal complexity however through local approaches, results have better quality.
the humans. Contrarily, finding a correct correspondence of several images can be a very elaborate task for an automatic method but humans performs it very well and fast. This method takes advantage of the qualities of the automatic and human behaviours since the automatic deduction of the pose is done by the automatic method conversely the human can help on deducting the point correspondences when the automatic method fails to do it.
Chapter 2

If a ‘religion’ is defined to be a system of ideas that contains unprovable statements, then Gödel taught us that mathematics is not only a religion, it is the only religion that can prove itself to be one.

John D. Barrow

2. The proposed model

Figure 1 shows a schematic view of our method based on a module that we have called interactive pose estimator and a human-machine interface\(^9\). In this example, cameras are embedded on the robots. The input of the general system is a set of 2D images and the output is their relative poses and the estimated errors (as the GPS does it). The human-machine interface receives the 2D images from the cameras, their current relative poses and the number of mapped points per pair of images from the interactive pose estimation module. The human-machine interface only outputs the user point-to-point mapping impositions to the interactive pose estimation module. Besides, the interactive pose estimation also receives the 2D images and then deducts and sends the relative poses estimation and the regression errors to the system that controls the robots.

The human-machine interface is as follows. On the left side of it, the user visualises the deducted current relative pose of the cameras (2D position on the land and robot orientation). On the middle of the interface, it is shown the

\(^9\) This scheme is a previews of the user interface showed in the last chapter.
The proposed model

number of mapped points between any combinations of cameras. The lowest values of the number of mappings are highlighted on bold to attract the attention of the user. On the right side of it, the user visualises the 2D images of two manually selected cameras together with the imposed point-to-point correspondences. The user can visualise any of the combinations of 2D images by selecting one of the cells in the middle of the human-machine interface or respectively robots. Then one can update the imposed correspondence by erasing or creating mappings between points. Two robots (or cameras) in the left panel and the pose errors in the central panel that correspond to the current images in the right panel are highlighted in red.

![Diagram](image)

Figure 1: Basic scheme of our method composed of our interactive pose estimation module and the human-machine interface.

Figure 2 schematically shows our method. We suppose there is a controlling system that needs to know the poses of some 2D cameras. As commented, given only the images from these cameras, our system deducts their poses together with an error and sends this information to the controlling system\(^\text{10}\).

The human-machine interface, which is explained partially previously and completely in the last chapter of this paper, shows the user the position of the

\(^{10}\) These three rows summarized in few words the whole system.
whole cameras and also the images that the user selects. It is a friendly interface to impose point-to-point mappings between pairs of images.

The first module of our system is the scheduler\(^1\) that selects which \(N\) cameras are going to be considered and returns their current images \(I_i, I_j, \ldots, I_k\). Several strategies can be used depending on the application. If the human does not interact on the human-machine interface, the strategy could be to select the cameras that the current number of mappings is the largest and also depending on the deducted distances between them. Nevertheless, when the human selects a pair of cameras, these ones are scheduled with the closest ones.

The next two steps are composed of salient point extractors and feature matching methods. Several methodologies summarised in the previous section can be used depending on the application. The first one generates a set of points per image, \(P_i, P_j, \ldots, P_k\) and the second one generates the \(N \cdot (N - 1)/2\) correspondences \(M_{i,j}, M_{i,k}, \ldots, M_{j,k}\) between these sets (we suppose \(M_{i,j} = M_{j,i}\)). The aim of the next module, called human-assisted matching estimator, is to adapt the automatically deducted set of point-to-point mappings to the mappings imposed by the human. We call the updated set of correspondences as \(M'_{i,j}, M'_{i,k}, \ldots, M'_{j,k}\). We explain this module in detail in the section above. The structure from motion module deducts the homographies \(H_{i,i}, H_{i,j}, \ldots, H_{i,k}\) in the 3D space that convert the pose of the first selected camera to the rest of the selected ones. To do so, it needs the input to be the set of point-to-point mappings in the 2D space. Several methodologies have been summarised in the previous section\(^12\).

The last module is the cooperative pose estimator. The aim of this module is to return the \(M^2\) 3D homographies between the whole set of cameras, \(H_{1,1}\),

---

\(^1\) Scheduler has important roles since it is a dynamic module that change his work depending on the system’s states. In fact when the system does not need to use cooperative method, the scheduler has the only task to take the images and put them in the system. Otherwise the scheduler became smarter in case the cooperative method is necessary selecting the appropriate images to put in the system.

\(^12\) We want to remark that the aim of this paper is not to discover new approaches to extract key points or to adapt them.
The proposed model

$H_{1,2,...,H_{M,M}}$ and their estimated error. On the one hand, each time the whole process is computed, only a sub-set of cameras are selected and therefore, only a sub-set of 3D homographies are deducted by the structure from motion module. On the other hand, if the point of view of some cameras is such that it is not possible to share any point in the 3D space, then, it is impossible the structure from motion module to deduct their respective 3D homographies.

\[ \text{Figure 2: Scheme of our human assisted and pose estimation method.} \]

2.1. Human-assisted matching estimator

Through the human-machine interface, the user can perform three different operations that we call: False Mapping, True Mapping and Set Mapping. It is usual the feature matching algorithms not only to return a set of mappings but also a goodness of each mapping. This goodness is related on the distance between the mapped features extracted at the points. Thus, these three operations influence on a specific point-to-point mapping and also on the goodness of this mapping. The user interacts on the human-machine interface in a non-synchronous way. For this reason, the human-assisted matching estimator can receive only one user operation or some of them. The formats of these operations are as follows.

- **False Mapping** ($p_i^a, p_j^b$): The user considers the mapping from point $a$ on image $i$ to point $b$ on image $j$ has to be deleted since it is not correct.
• *True Mapping* \((p^a_i, p^b_j)\): The user is completely sure that the mapping from point \(a\) on image \(i\) to point \(b\) on image \(j\) is correct.

• *Set Mapping* \((p^a_i, p^b_j)\): The user imposes a new mapping from point \(a\) on image \(i\) to point \(b\) on image \(j\).

When the human assisted matching estimator receives the action *False Mapping* \((p^a_i, p^b_j)\), it checks the existence of this mapping and, if it exists, deletes it.

Moreover, when this module receives the action *True Mapping* \((p^a_i, p^b_j)\), it checks the existence of this mapping and, if it exists, increases its goodness parameter to the maximum. Finally, action *Set Mapping* \((p^a_i, p^b_j)\) makes the module to impose this new mapping with the maximum goodness. This module has the *Matching Matrix*, or *MM* for short, in which there is the last mapping between a pair of images in each cell. The user impositions directly update this matrix. Figure 3 (left) represents three robots performing guiding tasks. Robots fence the visitor group to force them to follow a specific tour. Robots need to work in a cooperative manner to keep a triangular shape in which people have to be inside. In these cooperative tasks, it is crucial to have a low-level computer vision task so that images extracted from the three robots and some static cameras in the surrounding are properly aligned to correctly deduct their relative poses. In this environment, there is a human that, through our human-machine interface, gives orders to the robots and controls their tasks. What we propose in this paper is the human can also visualise the images of the cameras and interact in a low level task through asynchronously imposing salient points to salient points mapping. Figure 3 (right) shows the images taken from the first two robots and three mappings that the human has imposed.

Note the system deducts the poses automatically but, when the human

---

13 This could be a future aspects of our system since at the moment it works only comparing 2D images and not video.
The proposed model asynchronously imposes a mapping then it takes into consideration this mapping and it continues its process automatically\textsuperscript{14}.

**Figure 3:** A representation of three robots performing guiding tasks and images taken from the cameras on the first two robots.

Besides, we say it is a cooperative model since the relative pose of two robots is deducted through the other robots (in term of homographies) when these robots do not share any part of the scene they visualise. **Figure 4** shows the pose of three robots. Robot 1 and 2 can deduct their relative pose but this is not possible between robot 1 and 3 since they do not share any main part of their images. This problem is solved through the cooperation of the robots. Robot 2 deducts its relative pose with respect to robot 3 and shares this information\textsuperscript{15} with robot 1.

**Figure 4:** Three robots visualising the same scene.

\textsuperscript{14}In this case the system understand which feature are more important selecting the closer one and discarding the other.

\textsuperscript{15}This is a well-known operation especially in robotic and other correlate disciplines.
2.2. Cooperative pose estimator

The cooperative pose estimator receives a set of 3D homographies from the structure from motion module. As commented, the homographies in this set are only related on pairs of images selected by the scheduler. The cooperative pose estimator has an $M \times M$ matrix, that we call Direct Homographies or $DH$ for short. In each cell, there is the last computed homography by the structure from motion module: $DH[i,j] = H_{i,j}$. Then, this matrix is updated each time that cooperative pose estimator receives new homographies. Note the diagonal of this matrix is filled with the identity homography. When the application initialises, the whole matrix is empty and it is being filled when the scheduler begins to output images. Nevertheless, the cells that cameras do not share any point or too few, always remain empty. Since we wish to output the homographies between the whole cameras, we define an $M \times M$ matrix, that we call Computed Homographies or $CH$ for short. The non-empty cells in the $DH$ matrix are copied to this new matrix when new data is available but the rest of the cells are deducted in a cooperative manner. It is important to mark that cooperative method is not necessary in case the homography matrix is complete however even in this case, could be helpful as one more check point (in term of result’s quality). This approach should be evaluate case by case, since could be onerous and unnecessary.

Figure 5 shows a scene where five robots visualise an object and there is a high overlapping between (about 16 degrees) images of the first three robots but robot 4 only shares part of the image with robot 3 and robot 5. Initially, they do not know the relative position of the other robots since robots do not have any system to locate themselves in the scene and therefore they do not know which robot is the closest to another. Table 1 shows $DH$ matrix ($I$ represents the identity). Robot 1 can deduct the relative pose of robot 2 directly (through homography $H_{1,2}$) but it is not able to deduct the relative pose of robot 4 or
The proposed model

robot 5\textsuperscript{16}. Nevertheless, in the case of robot 4, this information can be indirectly estimated through robot 3 as the product of homographies $H_{1,4} = H_{1,3} \cdot H_{3,4}$. Another option could be to compute this homography so that robot 2 takes part of it, $H_{1,4} = H_{1,2} \cdot H_{2,3} \cdot H_{3,4}$. Since we do not know which is the best way to deduct the relative pose, we estimate the mean error is zero and we compute the empty cells in $CH$ through the average of angles and positions of the available homographies in $DH$.

\textbf{Figure 5: Five robots looking at an object.}

\textsuperscript{16}In some cases, the unmatched between images couldn’t be caused by the large angle degree between the images. Can happen that the key point extract focus on different part of the pictures.
Algorithm 1 returns $CH$ matrix and the estimated error of each homography given $DH$ matrix. This algorithm is executed by the cooperative pose estimator module each time new homographies are received from the structure from motion module. It is composed of two main steps. The first one updates the cells in $CH$ that the module has received new homographies. In the second step, the empty cells in $CH$ are filled in a cooperative way. The first time the Loop — While is executed, the mean is computed on the cases that there is only one extra camera. $H_{1,4}$ and $H_{2,4}$ can only be deducted through camera 3, $H_{1,4} = H_{1,3} \cdot H_{3,4}$ and $H_{2,4} = H_{2,3} \cdot H_{3,4}$. Moreover, $H_{3,5}$ can only be deducted through camera 4, $H_{3,5} = H_{3,4} \cdot H_{4,5}$. Figure 6 shows the new obtained homographies and Table 2 shows $CH$ after the first time the Loop — While is executed.

Table 1: Direct Homographies matrix, DH

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>I</td>
<td>$H_{12}$</td>
<td>$H_{13}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R2</td>
<td>$H_{21}$</td>
<td>I</td>
<td>$H_{23}$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R3</td>
<td>$H_{31}$</td>
<td>$H_{32}$</td>
<td>I</td>
<td>$H_{34}$</td>
<td>-</td>
</tr>
<tr>
<td>R4</td>
<td>-</td>
<td>-</td>
<td>$H_{43}$</td>
<td>I</td>
<td>$H_{45}$</td>
</tr>
<tr>
<td>R5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$H_{54}$</td>
<td>I</td>
</tr>
</tbody>
</table>

Figure 6: Homographies deducted in the first run of the loop in Algorithm 1
Table 2: Computed Homographies matrix (CH) deducted in the first run of the loop in Algorithm 1

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>I</td>
<td>H_{12}</td>
<td>H_{13}</td>
<td>H_{14}</td>
<td>-</td>
</tr>
<tr>
<td>R2</td>
<td>H_{21}</td>
<td>I</td>
<td>H_{23}</td>
<td>H_{24}</td>
<td>-</td>
</tr>
<tr>
<td>R3</td>
<td>H_{31}</td>
<td>H_{32}</td>
<td>I</td>
<td>H_{34}</td>
<td>H_{35}</td>
</tr>
<tr>
<td>R4</td>
<td>H_{41}</td>
<td>H_{42}</td>
<td>H_{43}</td>
<td>I</td>
<td>H_{45}</td>
</tr>
<tr>
<td>R5</td>
<td>-</td>
<td>-</td>
<td>H_{53}</td>
<td>H_{54}</td>
<td>I</td>
</tr>
</tbody>
</table>

Algorithm 1

Input DH
Output CH ERROR
N=length(DH);

%% Step one
for i=1:N
  for j=1:N
    if not(empty(DH[i,j])
      ERROR[i,j] = 0 % Error is 0 if directly computed
      CH[i,j] = DH[i,j]
    endif
  endfor
endfor

%% Step two
do
UPDATE=false
for i=1:N
  for j=1:N
    if empty(DH[i,j]) % New data does not have info.
      for k=1:N
        if not(empty(CH[i,k]) & not(empty(CH[k,j]))
          Hk=CH[i,k]*CH[k,j]
          Pk=Pose(Hk)
          UPDATE=true
        else
          Pk=empty
        endif
      endfor
      Pij=mean(P1,P2,..,Pn) % Empty Pij are discarded
      ERROR[i,j]=Standard Deviation(P1,P2,..,Pn)
      CH[i,j]=Homography(Pij) % Return empty if Pij is empty
    endif
  endfor
endfor
while UPDATE
End Algorithm
In the second run of the loop, the mean is computed on the cases that there are two extra cameras. For instance, $H_{1,5}$ can be computed as $H_{1,3} \cdot H_{3,5}$ but since $H_{3,5}$ was deducted in the first loop as $H_{3,4} \cdot H_{4,5}$ then in fact, we have that there are camera 3 and camera 4 involved, $H_{1,5} = H_{1,3} \cdot H_{3,4} \cdot H_{4,5}$. Nevertheless, another options is possible, which is $H_{1,5} = H_{1,4} \cdot H_{4,5}$. Therefore, $H_{1,5}$ is computed in the second run of the loop as the mean of the first option $H_{3,4} \cdot H_{4,5}$ and the second option $H_{1,4} \cdot H_{4,5}$.

The pose error is set to zero if the homography is directly computed by the structure from motion module or it is computed in the first run of the loop. In the other runs of the loop, the error is considered to be the standard deviation of the whole poses involved on computations the new homographies.

The algorithm stops when the whole cells are filled or it is not possible to fill them. This last case appears when there are two or more sets of cameras without shearing enough points.
3. Practical Validation

We propose three different databases$^{17}$ to validate our method that are summarised in table 1. In the following sub-sections, we explain each database and the errors committed while estimating the pose of the cameras. Each database is composed of a set of images and the ground truth of the 3D poses of each image.

Databases are available at:

(http://deim.urv.cat/~francesc.serratosa/databases/).

The feature point extractor has been SIFT (Lowe, D.G., 2004), the matching feature algorithm has been the Matlab function matchFeatures,

(http://es.mathworks.com/help/vision/ref/matchfeatures.html),

and the structure from motion method has been Bundler

(http://www.cs.cornell.edu/~snavely/bundler/).

$^{17}$ These three databases try to build application for different fields like pattern recognition, computer vision surveillance and robotic respectively described in the following section.
The Matlab code of our method is available at:

(http://deim.urv.cat/~francesc.serratosa/SW/).

Table 3: Mean features of the three used databases

<table>
<thead>
<tr>
<th>DATABASE</th>
<th># IMG</th>
<th># CAM</th>
<th>STATIC/MOBILE</th>
<th>IN/OUT DOOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAGRADA FAMILIA</td>
<td>360</td>
<td>8</td>
<td>Static</td>
<td>Outdoor</td>
</tr>
<tr>
<td>COURTYARD</td>
<td>32</td>
<td>2</td>
<td>Mobile</td>
<td>Outdoor</td>
</tr>
<tr>
<td>MIVIA LAB</td>
<td>5</td>
<td>5-2</td>
<td>Static</td>
<td>Indoor</td>
</tr>
</tbody>
</table>

3.1. First analysis: Camera positioning on open spaces

Sagrada Família database is a sequence of 360 2D-images manually taken around the Sagrada Família church in Barcelona (Spain) and looking at the centre of it. The average distance between two consecutive shots is 1.1 meters. Given the whole sequence, we used the method called Bundler (SNAVELY & TODOROVIC, 2011) to extract the pose of each camera that we consider is the ground truth. **Figure 7**: 3D points extracted by Bundler method of Sagrada Família database and the deducted 3D poses of the cameras shows the set of 3D points extracted from Bundler method and also the poses of the cameras. More than 100.000 3D points where extracted and for this reason, an I7 processor spent more than 7 hours to deduct the 3D poses of the cameras. The database is composed of 360 images, the 360 poses of the cameras and the set of 2D and 3D salient points. In this experiments, we have only used the set of images to deduct the poses and the cameras’ poses as a grunt truth.
Figure 7: 3D points extracted by Bundler method of Sagrada Familia database and the deducted 3D poses of the cameras.

Figure 8: Database Sagrada Familia Images shows the first eight images of this database. Figure 9 the deducted poses of the cameras given only these eight images and the set of points that Bundler has obtained and it considers that appear in the eight images. In this case, the runtime spent to compute the cameras’ poses is lower than a second. There is a building on the left hand of these images that appears to be represented as a cloud of points that has a linear shape in front of the cameras. Note the branches of a tree that hinders the proper extraction of salient points in the images.
Each result we present in this section is the average of 360 consecutive tests. In each test, we have taken 8 consecutive images\textsuperscript{18} (we suppose there are 8 cameras each time) from the 360 images that constitute the database with the angle of separation between images. For instance, test number 350 with angle of separation 2 is composed of images 350, 353, 356, 359, 2, 5, 8 and 11. Figure\textsuperscript{10} shows the relative position error (left) and the relative angle error (right) returned by the system with respect to the number of interactions and the angle of separation between images.

As it is supposed to be, far away are the images, larger is the error since less 2D-points are shared and also larger is the distortion between images. It is clear that with one interaction, the error is drastically reduced independently of the angle of separation between paired images. When more than one interactions were imposed, the error was only slightly reduced. Moreover, we have checked that with angles equal or larger than 5, the method was not able to deduct the poses without the cooperative module. This is because the angle between the

\textsuperscript{18} Choosing 8 images we have covered an angle of 10 degrees and for each degree of separation we double the coverage.
first camera and the eighth one is too large. For instance, with angle 5 the angle between the first camera and the last one is 35 and with an angle of 20, this last angle is 140.

![Figure 10: Mean errors on the positions in meters (left) and normalised angles (right) with respect to the number of interactions and the angle of separation between cameras.](image)

Given the results Figure 10, we could recommend to the user to interact a maximum of two times per pair of cameras through all pairs of cameras instead of interacting more than two times in some few pairs of cameras and keeping some pairs of cameras without interaction. From Figure 10, we conclude it is better to interact on the pairs of cameras that have the largest pose errors because in these cases the human interaction accentuates the pose errors to decrease.

### 3.2. Application 1: Robotics positioning

Courtyard database is composed of 32 images taken by two mobile robots (16 images per robot) in a courtyard of Universitat Politècnica de Catalunya. Figure 11 shows two of these images
Figure 11: Two images in the sequence.

Figure 12 (left) shows the distance between both robots given the 16 shots. In the first nine images, the distance where between 8 and 11 meters and from shot ten to thirteen robots get closer until being almost touching. In the last three shots, the robots separate again. Figure 12 (right) shows the error on the robots’ position without human interaction and with one human interaction given the 16 shots. In the sixteen shots, the human interactions make to decreases the position error. The position error not only depends on the distance between robots but also on the point of view\textsuperscript{19}. In the thirteen shot, robots are close to each other but they share few salient points. For this reason, the error is larger than the real distance.

\textsuperscript{19} As we explained previously, even if the two robot have the same point of you, few details involve big differences in the results.
3.3. Application 2: Camera calibration

This section contains the instruction set of the camera position and orientation, in order to acquire the images for our purpose (camera calibration). The environment that we consider is a simple room from Salerno University. In the centre of this room there will be a table and on it an object. We consider our first camera with coordinator (0x, 0y, 0z) as axis origin. In order to process the position and the orientation of the whole cameras system, we consider a camera as a 3D axis system like in Figure 13.

In the figure above, we also consider the Z-axis along the camera’s lent pointing the object on the table. Y-axis will point the ceiling and consequently X-axis will follow the right-hand rule. Each one of other camera will be consider in respect
of the first camera in term of position and orientation. Fixing one of three axis, the system may rotate around that axis. Considering that for the three axis, we will have three-rotation angles as in Figure 14.

3.3.1. Images acquisition

In order to obtain good results, we need at least a set of 16 images taking around the table in order to cover all the 360 degrees\(^{20}\). For testing our system, we consider a table as the centre of a circumference that we will divide in 16 parts. Therefor each picture will be taken in a range between 18 and 24 degrees.

\[ [18-24] \text{ degrees} \]

\(^{20}\) In case the images set is less than 16 images, the cooperative method is necessary to have all the cameras homographies.
For a better performance of our system, we need to use a 640x480-image resolution (or more) in an environment light-control. In this case we have enough key point to extract. However, if the resolution is too high, the system will take too much time and probably the processing will be stopped.

Following this section, we find a camera position’s simulation. In particular we suppose that we have two cameras, camera 1 and camera 2 with position (0,0,0) and (-X₂,+Y₂,+Z₂) respectively. We also supposed to change camera 2, in order to have different angles (+30°,+45°,0). In fact, camera 2 has two angles variation respect camera 1, the first one by X-axis and following by Y-axis. The position and orientation of camera 2 is respect to the position and orientation of camera 1 like in Figure 15 and Figure 16.

*Figure 15: Top View camera position*
3.3.2. Practical specifics

We need also to have information about:

- CCD width
- Resolution
- Focal length (pixels)

These parameters change the images views in term of distortion and other problems like fish-eye effect. In order to estimate the correct camera position, we need to know them.

It is also important to set the base room’s coordinator that we are considering, in this way, it fixed the position and angles of camera 1. As it shows in the Figure 17, we consider the centre of the table as the centre of the global axis and while camera 2, camera 3 and so on will refer to camera 1, and this last will refer to the global axis coordinator. All the angles and initial positions are defined. In Figure 18, as in the previously figure, it is possible to see in which way we
consider the distance between the camera 1 and the centre of table or object\textsuperscript{21}. Under such setting, we can omit the object’s size because it can be included in the distance along Z-axis (if camera 1 is collinear with it).

\textbullet\ Figure 17: Base coordinator for camera 1

\textbullet\ Figure 18: Side view of base coordinator for camera 1

\textsuperscript{21} In order to simplify the experiment, we consider the object as a point otherwise table’s center does not correspond with object’s center.
3.3.3. Salerno Lab Database

Salerno Lab database is a sequence of five images taken in a lab at Salerno University in accordance with the previous instruction set. The aim of this database is to calibrate the cameras to later perform surveillance and human recognition tasks. Figure 19 shows the five images and the real pose of the cameras.

![Figure 19: Salerno lab database composed of 5 images and the cameras’ poses](image)

We first tried to deduct the cameras’ poses with the five images and without human interaction. In the first round, the method was not able to deduct the homographies of the fifth camera with respect to cameras 1, 2 and 3 due to the large differences between the fifth image and the rest of the images (Table 4: Direct homographies matrix shows the direct homographies matrix). Nevertheless, the scheduler realised that the matching modules were able to deduct 45 mappings between the fourth and the fifth image (Table 5: Matching matrix shows the matching matrix). For this reason, the scheduler selected cameras 1, 2, 3 and 4 in a second round and selected cameras 4 and 5 in a third round (As showed in the same tables, the matching number decrease while the angle amplitude increase). Then, with the deducted homographies, the
cooperative pose estimator was able to deduct the whole homographies. Later, the user imposed a point mapping between the whole pairs of images.

*Table 4: Direct homographies matrix*

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>I</td>
<td>H_{12}</td>
<td>H_{13}</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C2</td>
<td>H_{21}</td>
<td>I</td>
<td>H_{23}</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C3</td>
<td>H_{31}</td>
<td>H_{32}</td>
<td>I</td>
<td>H_{34}</td>
<td>-</td>
</tr>
<tr>
<td>C4</td>
<td>-</td>
<td>-</td>
<td>H_{43}</td>
<td>I</td>
<td>H_{45}</td>
</tr>
<tr>
<td>C5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>H_{54}</td>
<td>I</td>
</tr>
</tbody>
</table>

*Table 5: Matching matrix*

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>-</td>
<td>75</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C2</td>
<td>75</td>
<td>-</td>
<td>66</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>C3</td>
<td>29</td>
<td>66</td>
<td>-</td>
<td>64</td>
<td>0</td>
</tr>
<tr>
<td>C4</td>
<td>0</td>
<td>17</td>
<td>64</td>
<td>-</td>
<td>45</td>
</tr>
<tr>
<td>C5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>45</td>
<td>-</td>
</tr>
</tbody>
</table>

*Figure 20:* Ground truth and deducted poses (with and with one human interaction). shows the real pose of the cameras and the deducted poses without interaction and with one human interaction. We can see that the quality of the poses increases with only one human interaction per pair of images\(^{22}\).

\(^{22}\) It is important to remark, that this is just one human impose. It is trivial to say that more human imposes involve more result’s quality however at one step, the difference between two consecutive results will be irrelevant for the system’s aim.
3.4. User Simulation

In this section we show how to use our application step by step, selecting the input for the main function and showing results by two main figures\(^2^3\). The first figure is just simple the scene top view while the second one, it is an interactive window that shows robots’ information as matched and coordinator points. Beginning the application, we should selected the dataset to process. If we want to create a new image set (button “NEW”) or add new images to the previous set (button “ADD”) as Figure 21.

\(^{2^3}\) In this case, the term “figure” is used to mean Matlab’s figure. Therefore in this kind of figure it is possible to interact and to make a user interface.
Whereas at the beginning image set is empty, as it’s written in the previous figure (Image Set: EMPTY), therefore pushing “NEW” and choosing the pictures that we are interest, we create the new dataset. In our case we chose four picture as follow:

![Image Set: EMPTY](kermit000.jpg kermit001.jpg kermit002.jpg kermit003.jpg)

The system asks if we are sure about our selection and if we want to choose some other pictures. Following the instruction, we complete the image acquirement part. In the next step, the system asks to choose an “image Base”, this mean to choose one of the images in the new set to be the one where the other refer to it\(^\text{24}\). For example if we are sure that in the first image, some point appear also in the other images, the first image is the good candidate to be

\(^{24}\text{This step is not mandatory. The system is able to recognize which pictures as the best matching with the other ones. Usually, as you can imagine, it is the middle one.}\)
selected for basic image. Chose the basic image, the system asks to choose a point that we think appears in all the images like in Figure 22.

Selected the point in the Image Base, the system asks to choose the corresponding point in the other images as it shows in Figure 23.

Figure 22: Image Base

---

In our case, we have selected the first image as Image Base because there is not a big angle variation between images therefore we are sure that a point in the middle appears in the first image appear also in the other images. Figure 23, after selected the point, will ask a confirmation of the choice. Finally, it will update.
After chose all the corresponding point in the other picture, we have completed the system input. The system for each step requires a confirmation. In the next stage, the system process as in Figure 2, therefor it takes the whole image set and starts to extract the salient point by SIFT method explained previously. This part takes about 1 second per picture, it depends by resolution and images total. At this point, the key matcher begins to create the matching between all the salient points. Same reasoning about timing. When the system has all the matched points, the structure for motion (Bundler’s method) begins to process all the results. The output it goes to Cooperative method and Human Impose blocks\textsuperscript{26}. Both try to rectify the main output as cameras’ homographies, 3D points and other information. Structure from motion, human impose and cooperative method blocks are the faster one since it is just a matrix matching operation. As we told at the beginning of this section, the output of our application is two figures. The first one in Figure 24, it shows the scene top view. This is just an information without any interaction that can be useful to figure the system process. In fact if we compare the pictures in our image set with the

\textsuperscript{26} Some output showed in Figure 2, are internal and other are to process distance error in case we have the ground truth. For this reason the output showed to the user is just a partially output that change in accordance with user aim.
scene top view we can easily deduct salient part. For example on the right of the 4-cameras (in green colour), there is a heap of points (in red colour) that show the billboard. In addition to this, it is possible to see the Sagrada Familia corner in the camera direction. The second figure, contains three main system features. The first one, it is interactive and define the information of the other ones. In Figure 25, on the left there is a simple zoom top view of cameras' position. As the title suggest, that is an interactive window where is possible to select two robots. Choosing that, the system shows two additional information. The first one (in the centre), it is the matrix of matching, which shows the total points that are matched between two robots. As the picture shows, this is a symmetric matric, changing the robots selected it involves the motion of the red squares inside the matrix. On the right, there are the first five main key points who appear in both images (i.e. Robot 4 and Robot 2)

![Figure 24: Top view system output](image)

---

27 In this scene top view, it is easy to see how much noise take part in our system. Noise depends especially on the dataset used.
Figure 25: Main System Output
Conclusion

We have presented an interactive and cooperative method to deduct the poses of some cameras given only the 2D images taken from these cameras. It is composed of any algorithm that extracts salient points, aligns them and performs structure from motion to deduct the cameras’ poses. Moreover, we have added two new modules. The first one permits to deduct the pose of the cameras that do not share any point through a cooperative algorithm. The second one adds interactivity to the system allowing the user to impose some point-to-point mappings when one believes it is needed to increase the pose accuracy.

In the experimental section, we have seen that, in some cases, the structure from motion module is not able to deduct the pose of the cameras but when the cooperative method is used, these poses can be estimated. Besides, in the extreme cases that the structure from motion plus the cooperative module is not able to deduct the pose or its error is really high, it is worth letting the human to interact by imposing some point-to-point mappings.

The system is being currently used in a large project, in which social robots guide people through urban areas. Usually, there is a human supervisor that interacts on the fleet of robots and sends them orders. In some cases, robots have to stop since they lose their pose. In these cases, the human interacts by simply mapping pairs of points and the robots recover their pose and therefore they can continue their high level task.

As a future work, we want to study in depth the impact of using our method on these high level tasks and how frequent the interaction is needed. Moreover, we will study their behaviour in more difficult frameworks, such as, low image quality or several mobile objects interfering in the robots’ task.

The first step will be working on videos instead images input. For this reason an appropriate user interface will be required in order to allow the user to work on videos.
A second step will be updating the human impose input. In fact, thinking to have over 10 images per input makes the user spend time to select a point for each images. For this reason, making the system automatic or semi-automatic (meaning that system can suggest a matching under an appropriate threshold moreover avoid a user selection in case the match is over an appropriate threshold) could be an advantage for the user interaction and system performance.

Concluding, a good step forward, will be installing this system on a real robot. Robot NAO from Aldebaran Company can be a good candidate for his adaptation to the system languages (Matlab and C++).
Figures

Figure 1: Basic scheme of our method composed of our interactive pose estimation module and the human-machine interface. .......................................................... 11

Figure 2: Scheme of our human assisted and pose estimation method. ............ 13

Figure 3: A representation of three robots performing guiding tasks and images taken from the cameras on the first two robots....................................................... 15

Figure 4: Three robots visualising the same scene. ......................................... 15

Figure 5: Five robots looking at an object.......................................................... 17

Figure 6: Homographies deducted in the first run of the loop in Algorithm 1..... 18

Figure 7: 3D points extracted by Bundler method of Sagrada Familia database and the deducted 3D poses of the cameras .................................................. 23

Figure 8: Database Sagrada Familia Images...................................................... 23

Figure 9: The first eight images of Sagrada Familia database and the deducted 3D points and poses of the cameras ................................................................. 24

Figure 10: Mean errors on the positions in meters (left) and normalised angles (right) with respect to the number of interactions and the angle of separation between cameras. ........................................................................ 25

Figure 11: Two images in the sequence.............................................................. 26

Figure 12: Distance between both robots and position errors throughout the 16 shots. ........................................................................................................ 26

Figure 13: Camera Definition ........................................................................... 27

Figure 14: Non-positive rotation around X-axis ............................................... 28

Figure 15: Top View camera position................................................................. 29

Figure 16: Side View camera position............................................................... 30

Figure 17: Base coordinator for camera 1 ......................................................... 31

Figure 18: Side view of base coordinator for camera 1 .................................... 31

Figure 19: Salerno lab database composed of 5 images and the cameras’ poses 32

Figure 20: Ground truth and deducted poses (with and with one human interaction). ........................................................................................................ 34

Figure 21: Panel to create or update an image set ............................................. 35

Figure 22: Image Base ....................................................................................... 36
Figure 23: Corresponding point in the other picture ........................................ 37
Figure 24: Top view system output .................................................................... 38
Tables

Table 1: Direct Homographies matrix, DH ............................................................... 18
Table 2: Computed Homographies matrix (CH) deducted in the first run of the loop in Algorithm 1 ........................................................................................................... 19
Table 3: Mean features of the three used databases .............................................. 22
Table 4: Direct homographies matrix ..................................................................... 33
Table 5: Matching matrix ....................................................................................... 33
Bibliography


RUBIO A. (2015). EFFICIENT MONOCULAR POSE ESTIMATION FOR COMPLEX 3D MODELS, INTERNATIONAL CONGRESS ON ROBOTICS AND AUTOMATION.


UNHELKAR, SHAH, (2015). CHALLENGES IN DEVELOPING A COLLABORATIVE ROBOTIC ASSISTANT FOR AUTOMOTIVE ASSEMBLY LINES, EXTENDED ABSTRACTS OF THE TENTH ANNUAL ACM/IEEE - 19 -
INTERNATIONAL CONFERENCE ON HUMAN-ROBOT INTERACTION EXTENDED ABSTRACTS, pp: 239-240.

